



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년03월10일
(11) 등록번호 10-2226693
(24) 등록일자 2021년03월05일

(51) 국제특허분류(Int. Cl.)
H04N 19/53 (2014.01) H04N 19/105 (2014.01)
H04N 19/176 (2014.01) H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/53 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2019-0106202
(22) 출원일자 2019년08월29일
심사청구일자 2019년08월29일
(56) 선행기술조사문헌
KR1020190038371 A*
Jianle Chen, et. al., "Algorithm description for versatile Video coding and test model5", JVET of ITU-T SG19 WP3 and ISO/IEC JTC 1/SC 29/ WG 11, 14th meeting, 19-27 March, 2019, JVET-N1002_v2.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
강계원
서울특별시 마포구 독막로 145, 104동 904호(창전동, 서강쌍용에가아파트)
박상효
서울특별시 강서구 강서로5라길 77, 403호(화곡동, 자라하우스)
(74) 대리인
특허법인(유한)아이시스

전체 청구항 수 : 총 13 항

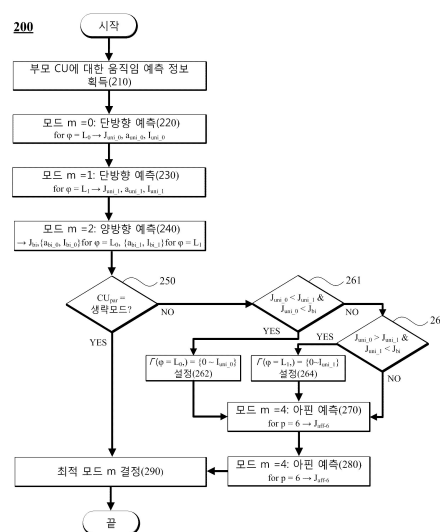
심사관 : 박상철

(54) 발명의 명칭 영상에 대한 고속 움직임 예측 방법 및 장치

(57) 요약

영상에 대한 고속 움직임 예측 방법은 인코더가 프레임의 특정 블록을 MTT(multi-type tree) 구조를 기준으로 부모 CU(coding unit)를 복수의 CU들로 분할하는 단계, 상기 인코더가 상기 부모 CU에 대한 움직임 예측 모드가 생략 모드인지 확인하는 단계 및 상기 인코더가 상기 복수의 CU들 중 어느 하나인 타깃 CU에 대한 움직임 예측을 하는 단계를 포함한다. 상기 인코더는 상기 부모 CU의 움직임 예측이 생략된 경우, 상기 타깃 CU에 대한 아핀 움직임 예측을 생략하고, 상기 인코더는 상기 부모 CU의 움직임 예측이 수행된 경우, 상기 타깃 CU에 대한 정규 움직임 예측의 정보를 기준으로 상기 타깃 CU의 움직임 예측을 위한 참조 프레임 범위를 한정하여 상기 타깃 CU에 대한 아핀 움직임 예측을 수행한다.

대표도 - 도4



(52) CPC특허분류

H04N 19/176 (2015.01)

H04N 19/70 (2015.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711081006
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기술진흥센터
연구사업명	정보통신 · 방송 연구개발사업-차세대(UHD)방송서비스활성화기술개발사업
연구과제명	6DoF지원 초고화질 몰입형 비디오의 압축 및 전송 핵심 기술 개발
기 여 율	1/1
과제수행기관명	이화여자대학교 산학협력단
연구기간	2019.01.01 ~ 2019.12.31

명세서

청구범위

청구항 1

인코더가 프레임의 특정 블록을 MTT(multi-type tree) 구조를 기준으로 부모 CU(coding unit)를 복수의 CU들로 분할하는 단계;

상기 인코더가 상기 부모 CU에 대한 움직임 예측 모드가 생략 모드인지 확인하는 단계; 및

상기 인코더가 상기 복수의 CU들 중 어느 하나인 타깃 CU에 대한 움직임 예측을 하는 단계를 포함하되,

상기 타깃 CU에 대한 움직임 예측 과정은

상기 타깃 CU에 대한 정규 움직임 예측을 수행하는 단계; 및

필요한 경우 상기 타깃 CU에 대한 아핀 움직임 예측을 수행하는 단계를 포함하되,

(i) 상기 인코더는 상기 부모 CU의 움직임 예측이 생략된 경우, 상기 타깃 CU에 대한 아핀 움직임 예측을 생략하고,

(ii) 상기 인코더는 상기 부모 CU의 움직임 예측이 수행된 경우, 참조 프레임을 사용하여 상기 타깃 CU에 대한 아핀 움직임 예측을 수행하고, 상기 참조 프레임은 상기 타깃 CU에 대한 정규 움직임 예측의 정보를 기준으로 범위가 한정되는 영상에 대한 고속 움직임 예측 방법.

청구항 2

제1항에 있어서,

상기 타깃 CU에 대한 정규 움직임 예측에서 단방향 예측의 최적 RD 비용이 양방향 예측의 최적 RD 비용보다 작은 경우,

상기 인코더는 상기 타깃 CU에 대한 정규 움직임 예측 중 최적 RD 비용이 최소인 단방향 예측에서 사용된 참조 프레임 리스트를 사용하여 상기 타깃 CU의 아핀 움직임 예측을 하되,

상기 참조 프레임 리스트의 범위 상한을 상기 타깃 CU에 대한 정규 움직임 예측에서 결정된 최적 참조 프레임 인덱스로 설정하여 상기 타깃 CU의 아핀 움직임 예측을 하는 영상에 대한 고속 움직임 예측 방법.

청구항 3

인코더가 프레임의 특정 블록을 트리 구조를 기반으로 상위 블록과 상기 상위 블록의 하위 블록으로 분할하는 단계;

상기 인코더가 상기 상위 블록에 대한 움직임 예측을 수행하여 움직임 예측 정보를 획득하는 단계;

상기 인코더가 상기 하위 블록에 대한 정규 움직임 예측을 수행하는 단계; 및

상기 인코더가 상기 하위 블록에 대한 아핀 움직임 예측을 수행하는 단계를 포함하되,

상기 하위 블록에 대한 아핀 움직임 예측 과정은

상기 상위 블록의 움직임 예측이 생략된 경우, 상기 하위 블록에 대한 아핀 움직임 예측을 생략하고,

상기 상위 블록의 움직임 예측이 수행된 경우, 참조 프레임을 사용하여 상기 하위 블록에 대한 아핀 움직임 예측을 수행하고, 상기 참조 프레임은 상기 하위 블록에 대한 정규 움직임 예측의 정보를 기준으로 범위가 한정되는 영상에 대한 고속 움직임 예측 방법.

청구항 4

제3항에 있어서,

상기 인코더는 VVC (Versatile Video coding) 영상에 대한 움직임 예측을 수행하는 영상에 대한 고속 움직임 예측 방법.

청구항 5

제3항에 있어서,

상기 인코더는 QT(quad-tree), TT(ternary tree) 및 BT(binary tree) 타입을 포함하는 MTT(multi-type tree) 구조를 기준으로 상기 특정 블록을 분할하는 영상에 대한 고속 움직임 예측 방법.

청구항 6

제3항에 있어서,

상기 정규 움직임 예측의 정보는

제1 참조 프레임 리스트를 이용한 단방향 예측의 RD 비용값(J_{uni_0}), 상기 제1 참조 프레임 리스트를 이용한 단방향 예측의 최적 참조 프레임 인덱스(I_{uni_0}), 제2 참조 프레임 리스트를 이용한 단방향 예측의 RD 비용값(J_{uni_1}), 상기 제2 참조 프레임 리스트를 이용한 단방향 예측의 최적 참조 프레임 인덱스(I_{uni_1}) 및 상기 제1 참조 프레임 리스트와 제2 참조 프레임 리스트를 이용한 양방향 예측의 RD 비용값(J_{bi})을 포함하는 영상에 대한 고속 움직임 예측 방법.

청구항 7

제6항에 있어서,

상기 인코더는 $J_{uni_0} < J_{uni_1}$ 이고 $J_{uni_0} < J_{bi}$ 이고,

상기 하위 블록에 대한 정규 예측에서 상기 제1 참조 프레임 리스트를 이용한 RD 비용값이 가장 작은 경우, 상기 제1 참조 프레임 리스트 범위를 "인덱스 0 ~ I_{uni_0} " 범위로 한정하여 상기 하위 블록에 대한 아핀 움직임 예측을 하는 영상에 대한 고속 움직임 예측 방법.

청구항 8

제6항에 있어서,

상기 인코더는 $J_{uni_0} > J_{uni_1}$ 이고 $J_{uni_1} < J_{bi}$ 이고,

상기 하위 블록에 대한 정규 예측에서 상기 제2 참조 프레임 리스트를 이용한 RD 비용값이 가장 작은 경우, 상기 제2 참조 프레임 리스트 범위를 "인덱스 0 ~ I_{uni_1} " 한정하여 상기 하위 블록에 대한 아핀 움직임 예측을 하는 영상에 대한 고속 움직임 예측 방법.

청구항 9

프레임의 특정 블록을 트리 구조를 기반으로 상위 블록과 상기 상위 블록의 하위 블록으로 분할한 정보, 상기 상위 블록에 대한 움직임 예측 정보 및 상기 하위 블록에 대한 움직임 예측을 수행하기 위한 코드를 저장하는 저장 장치; 및

상기 코드를 실행하여 상기 하위 블록에 대한 움직임 예측을 수행하되, 상기 하위 블록에 대한 움직임 예측은 정규 움직임 예측 과정과 필요한 경우 수행하는 아핀 예측 과정을 포함하는 프로세서를 포함하되,

상기 프로세서는 상기 상위 블록의 움직임 예측이 생략된 경우, 상기 하위 블록에 대한 아핀 움직임 예측을 생략하고,

상기 상위 블록의 움직임 예측이 수행된 경우, 참조 프레임을 사용하여 상기 하위 블록에 대한 아핀 움직임 예측을 수행하고, 상기 참조 프레임은 상기 하위 블록에 대한 정규 움직임 예측의 정보를 기준으로 범위가 한정되는 고속 움직임 예측 장치.

청구항 10

제9항에 있어서,

상기 프로세서는 VVC (Versatile Video coding) 영상에 대한 움직임 예측을 수행하는 고속 움직임 예측 장치.

청구항 11

제9항에 있어서,

상기 프로세서는 QT(quad-tree), TT(ternary tree) 및 BT(binary tree) 타입을 포함하는 MTT(multi-type tree) 구조를 기준으로 상기 특정 블록을 분할하는 고속 움직임 예측 장치.

청구항 12

제9항에 있어서,

상기 정규 움직임 예측의 정보는

제1 참조 프레임 리스트를 이용한 단방향 예측의 RD 비용값(J_{uni_0}), 상기 제1 참조 프레임 리스트를 이용한 단방향 예측의 최적 참조 프레임 인덱스(I_{uni_0}), 제2 참조 프레임 리스트를 이용한 단방향 예측의 RD 비용값(J_{uni_1}), 상기 제2 참조 프레임 리스트를 이용한 단방향 예측의 최적 참조 프레임 인덱스(I_{uni_1}) 및 상기 제1 참조 프레임 리스트와 제2 참조 프레임 리스트를 이용한 양방향 예측의 RD 비용값(J_{bi})을 포함하는 고속 움직임 예측 장치.

청구항 13

제9항에 있어서,

상기 프로세서는

상기 하위 블록에 대한 정규 움직임 예측에서 단방향 예측의 최적 RD 비용이 양방향 예측의 최적 RD 비용보다 작은 경우,

상기 하위 블록에 대한 정규 움직임 예측 중 최적 RD 비용이 최소인 단방향 예측에서 사용된 참조 프레임 리스트를 사용하여 상기 하위 블록의 아핀 움직임 예측을 하되,

상기 참조 프레임 리스트의 범위 상한을 상기 하위 블록에 대한 정규 움직임 예측에서 결정된 최적 참조 프레임 인덱스로 설정하여 상기 하위 블록의 아핀 움직임 예측을 하는 고속 움직임 예측 장치.

발명의 설명

기술 분야

[0001] 이하 설명하는 기술은 영상 코딩 기법에 관한 것이다. 특히 이하 설명하는 기술은 영상의 움직임 예측 기법에 관한 것이다.

배경 기술

[0002] 차세대 비디오 부호화 (Future Video Coding) 표준을 위하여 구성된 ITU-T SC16 WP3와 ISO/IEC JTC 1/SC 29/WG 11의 협력팀인 JVET(Joint Video Experts Team)은 HEVC(High Efficiency Video Coding)/H.265 보다 2배 이상의 부호화 성능을 갖는 VVC (Versatile Video coding)에 대한 표준화 작업을 진행 중이다. 최근 VCC 테스트 모델 (VTM)은 동일한 품질에 대하여 HEVC 보다 31% 가량 비트가 절감되는 성능을 보였다.

선행기술문헌

특허문헌

[0003] (특허문헌 0001) 미국등록특허 US 9866863 (2018.01.09)

발명의 내용

해결하려는 과제

- [0004] HEVC 영상에 대한 움직임 예측 분야에서도, 움직임 예측의 속도를 향상시키기 위한 연구들이 있었다. 종래 기술은 주로 HEVC의 구조 및 정보를 이용하여 움직임 예측의 속도를 높이는 것이었다. 그러나 VVC는 HEVC와는 다른 구조를 갖고, PU(prediction unit)과 같은 정보가 없기 때문에, 종래 기술을 적용하여 움직임 예측을 빠르게 수행할 수 없다.
- [0005] 이하 설명하는 기술은 VVC 코딩 과정에서 움직임 예측을 고속으로 수행하는 방법을 제공하고자 한다.

과제의 해결 수단

- [0006] 영상에 대한 고속 움직임 예측 방법은 인코더가 프레임의 특정 블록을 MTT(multi-type tree) 구조를 기준으로 부모 CU(coding unit)를 복수의 CU들로 분할하는 단계, 상기 인코더가 상기 부모 CU에 대한 움직임 예측 모드와 생략 모드인지 확인하는 단계 및 상기 인코더가 상기 복수의 CU들 중 어느 하나인 타깃 CU에 대한 움직임 예측을 하는 단계를 포함한다. 상기 타깃 CU에 대한 움직임 예측 과정은 상기 타깃 CU에 대한 정규 움직임 예측을 수행하는 단계 및 필요한 경우 상기 타깃 CU에 대한 아핀 움직임 예측을 수행하는 단계를 포함한다. (i) 상기 인코더는 상기 부모 CU의 움직임 예측이 생략된 경우, 상기 타깃 CU에 대한 아핀 움직임 예측을 생략하고, (ii) 상기 인코더는 상기 부모 CU의 움직임 예측이 수행된 경우, 상기 타깃 CU에 대한 정규 움직임 예측의 정보를 기준으로 상기 타깃 CU의 움직임 예측을 위한 참조 프레임 범위를 한정하여 상기 타깃 CU에 대한 아핀 움직임 예측을 수행한다.
- [0007] 고속 움직임 예측 장치는 프레임의 특정 블록을 트리 구조를 기반으로 상위 블록과 상기 상위 블록의 하위 블록으로 분할한 정보, 상기 상위 블록에 대한 움직임 예측 정보 및 상기 하위 블록에 대한 움직임 예측을 수행하기 위한 코드를 저장하는 저장 장치 및 상기 코드를 실행하여 상기 하위 블록에 대한 움직임 예측을 수행하되, 상기 하위 블록에 대한 움직임 예측은 정규 움직임 예측 과정과 필요한 경우 수행하는 아핀 예측 과정을 포함하는 프로세서를 포함한다. 상기 프로세서는 상기 상위 블록의 움직임 예측이 생략된 경우, 상기 하위 블록에 대한 아핀 움직임 예측을 생략하고, 상기 상위 블록의 움직임 예측이 수행된 경우, 상기 하위 블록에 대한 정규 움직임 예측의 정보를 기준으로 상기 하위 블록의 움직임 예측을 위한 참조 프레임 범위를 한정하여 상기 하위 블록에 대한 아핀 움직임 예측한다.

발명의 효과

- [0008] 이하 설명하는 기술은 이전에 코딩한 정보를 이용하여 하위 레벨의 블록에 대한 움직임 예측을 보다 빠르게 수행하게 한다. 이하 설명하는 기술은 정규 예측 과정 후에 수행하는 아핀 예측 과정을 생략하거나, 단축하여 전체 움직임 예측의 복잡도를 낮춘다. 이하 설명하는 기술은 HEVC 보다 높은 복잡도를 갖는 VVC 영상도 모바일 기기와 같은 한정된 성능을 갖는 장비에서 실시간 영상 녹화를 가능하게 한다.

도면의 간단한 설명

- [0009] 도 1은 VVC의 MTT(Multi-Type Tree) 구조에 대한 예이다.
- 도 2는 아핀 예측에 대한 예이다.
- 도 3은 VVC에서 CU에 대한 움직임 예측을 수행하는 과정에 대한 예이다.
- 도 4는 VVC에서 CU에 대한 고속 움직임 예측을 수행하는 과정에 대한 예이다.
- 도 5는 VVC에서 움직임 예측을 수행하는 인코더 장치의 구성을 도시한 예이다.

발명을 실시하기 위한 구체적인 내용

- [0010] 이하 설명하는 기술은 다양한 변형을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

- [0011] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0012] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 실시된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.
- [0013] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.
- [0014] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0016] 도 1은 VVC의 MTT(Multi-Type Tree) 구조에 대한 예이다.
- [0017] MTT는 VVC에 적용된 새로운 트리 구조이다. MTT는 종래 QT(Quad Tree) 외에도 BT(Binary Tree) 및 TT(Ternary Tree)를 사용한다. MTT는 하나의 블록을 QT, BT 또는 TT를 사용하여 분할할 수 있다. MTT는 하나의 CU(Coding Unit)를 다양한 서브 트리(sub-tree)로 분할할 수 있게 한다. 특히, BT와 TT는 QT의 하위에서만 존재할 수 있다. BT는 BT 또는 TT를 서브 트리로 둘 수 있다. 또 TT는 TT 또는 BT를 서브 트리로 둘 수 있다. BT 및 TT는 도 1과 같이 가로 방향(horizontal)과 세로 방향(vertical)으로 분할 가능하다. 트리 최대 깊이와 최소 블록 사이즈에 따라 하나의 블록은 보다 더 다양한 형태로 분할될 수 있다. 즉, VCC는 다양한 트리 구조를 사용하여 블록 분할을 다양한 형태로 할 수 있다. VCC는 다양한 트리 구조를 제공하는 MTT를 이용하여 향상된 코딩 효율을 제공한다.
- [0019] VVC는 아핀 움직임 예측/보상(affine motion estimation/compensation)를 사용한다. VVC는 HEVC와 달리 아핀 움직임 예측/보상 기술을 사용하여 블록의 다양성을 제공하고 압축 효율을 높인다.
- [0020] 도 2는 아핀 예측에 대한 예이다. 아핀 기술은 하나의 CU를 다양한 모양의 블록으로 예측할 수 있다. 특히, 아핀 변환은 움직임 벡터(motion vector)를 생성하기 위한 2가지의 부호화/복호화 기법(2가지 모델)이 존재한다. 아핀 변환의 부호화/복호화 기법의 종류는 파라미터의 개수로 결정된다.
- [0021] 두 가지 모델 중 하나는 도 2(A)에 도시한 4-파라미터 기반 아핀 변환이다. 다른 하나는 도 2(B)에 도시한 6-파라미터 기반 아핀 변환이다. 4-파라미터 기반 아핀 변환은 a 및 b의 두 가지 제어 포인트(control point)만 필요한 반면, 6-파라미터 기반 아핀 변환은 a, b 및 c의 세 가지 제어 포인트가 필요하다.
- [0022] 4-파라미터 기반 아핀 변환을 사용할 경우, 움직임 벡터는 아래의 수학식 1로 유도된다. 6-파라미터 기반 아핀 변환을 사용할 경우, 움직임 벡터는 아래의 수학식 2로 유도된다.
- [0023] 수학식에서 mv는 2차원 직교좌표계에서의 움직임 벡터를 나타낸다. mv는 (x,y) 위치에서 (mv^h, mv^v) 로 나타낼 수 있다. mv^h 는 x축 상의 지점이고, mv^v 는 y축 상의 지점이다.
- [0024] 도 2(A)와 같이 두 개의 벡터가 있고, 하나의 벡터 (a^h, a^v) 가 좌측 상단 코너에 있고, 다른 벡터 (b^h, b^v) 가 우측 상단 코너에 있다면, (x,y) 지점에 대한 mv는 아래 수학식 1과 같이 나타낼 수 있다.

수학식 1

$$\begin{cases} mv^h = \frac{b^h - a^h}{w}x + \frac{b^v - a^v}{w}y + a^h \\ mv^v = \frac{b^v - a^v}{w}x + \frac{b^h - a^h}{w}y + a^v \end{cases}$$

수학식 1에서 w 는 블록의 너비(width)이다.

전술한 2개의 벡터 외에 추가적인 벡터 (c^h , c^v)가 좌측 하단 코너에 있는 경우, (x, y) 지점에 대한 mv 는 아래 수학식 2와 같이 나타낼 수 있다.

수학식 2

$$\begin{cases} mv^h = \frac{b^h - a^h}{w}x + \frac{c^h - a^h}{h}y + a^h \\ mv^v = \frac{b^v - a^v}{w}x + \frac{c^v - a^v}{h}y + a^v \end{cases}$$

수학식 2에서 h 는 블록의 높이(height)이다.

이하 설명의 편의를 위하여 VTM 또는 인코더가 움직임 예측을 수행한다고 설명한다. VTM은 영상을 코딩하는 구성을 의미한다. 인코더는 영상을 인코딩하는 장치를 말한다. 물리적으로 PC, 서버 등과 같은 컴퓨터 장치가 움직임 예측을 수행한다.

아핀 예측은 단방향 예측(uni-prediction)과 쌍방향 예측(bi-prediction)을 수행해야 한다. 전술한 4-파라미터 기반 아핀 예측과 6-파라미터 기반 아핀 예측 모두 두 가지 방향성의 예측을 수행한다. 따라서, VVC에서 움직임 예측 과정에서 필요한 참조 프레임(reference frame)의 개수는 HEVC에 비하여 두배 이상이다. 움직임 예측의 복잡도가 높기 때문에, VTM은 어떤 블록에 대하여 움직임 예측을 수행할지 여부를 결정하기 위한 기준(임계값)을 사용할 수 있다. 이하 설명의 편의를 위하여, 종래 움직임 예측에서의 최적 율-왜곡(rate distortion, RD) 비용을 J_{CME} 라고 하고, VVC에서 4-파라미터 기반 움직임 예측의 최적 RD 비용을 J_{4-AME} 라고 명명한다. VTM은 J_{CME} 와 J_{4-AME} 를 특정 기준과 비교하여 6-파라미터 기반 움직임 예측의 생략(skip) 여부를 결정할 수 있다.

RA(random access) 설정하에서 인코더는 가용한 복수의 참조 프레임 중에서 블록의 RD 비용을 최소화할 수 있는 최적 참조 프레임을 결정한다. RD 비용 J 는 아래 수학식 3이 나타내는 라그랑주 승수법(Lagrangian Multiplier)을 사용하여 연산될 수 있다.

수학식 3

$$J = D + \lambda \cdot R$$

D 는 블록 왜곡(이미지 품질)을 나타내고, λ 는 라그랑주 승수를 나타내고, R 은 블록 비트(또는 추정 비트)를 나타낸다.

두 개의 참조 프레임 리스트들 L_0 및 L_1 이 사용되기 때문에, 단방향 움직임 예측 과정은 두 개의 리스트에서 가용한 모든 프레임들을 생성한다. 참조 프레임 리스트 L_0 은 제1 참조 프레임 리스트라고 명명할 수 있다. 참조 프레임 리스트 L_1 은 제2 참조 프레임 리스트라고 명명할 수 있다. 참조 프레임 세트를 Γ , 참조 프레임 리스트 $\varnothing$ 에서 가용한 모든 프레임들 세트를 $\Gamma(\varnothing)$ 라고 한다. 참조 프레임 i 를 사용하여 어떤 블록의 최소 RD 비용을 획득하는 과정은 아래 수학식과 같이 표현할 수 있다. 수학식 4는 단방향 움직임 예측에 대한 최소 RD 비용이고,

수학식 5는 양방향 움직임 예측에 대한 최소 RD 비용이다.

수학식 4

$$\operatorname{argmin}_{i \in \Gamma(\varphi) | \varphi=L_0, L_1} \{J(\varphi(i))\}$$

수학식 5

$$\operatorname{argmin}_{i \in \Gamma(\varphi) | \varphi=L_0, L_1} \left\{ \frac{J(\varphi(i) | \varphi=L_0)}{2} + \frac{J(\varphi(i) | \varphi=L_1)}{2} \right\}$$

Φ 는 모든 참조 프레임 리스트 세트를 나타낸다.

도 3은 VVC에서 CU에 대한 움직임 예측을 수행하는 과정(100)에 대한 예이다.

참조 프레임 리스트들은 L_0 및 L_1 이 사용된다고 전제한다. 전체 움직임 예측 과정을 간략하게 설명하면, 인코더가 MTT 노드인 하나의 CU에 대하여 동일한 참조 프레임 세트를 사용하여 전통적인 움직임 예측과 아핀 움직임 예측을 순차적으로 수행한다. 인코더는 모든 움직임 예측에 대한 RD 비용 J 을 결정하여, 최적 모드 m 을 결정한다. 인코더는 최적 모드 m 에 대하여 최적 움직임 벡터 a , 최적 참조 프레임 인덱스 i 를 얻는다.

$\varphi \in \{L_0, L_1\}$ 이고, $\Gamma(\varphi) = \{0 \sim n(\Gamma(\varphi)-1)\}$ 이다.

인코더는 먼저 정규 움직임 예측을 수행한다(110~130).

인코더는 L_0 를 사용하여 단방향 움직임 예측을 수행하고, RD 비용을 연산한다(110). 모드 $m = 0$ 은 L_0 를 사용하는 단방향 예측을 의미한다. 인코더는 $m = 0$ 에서 L_0 를 이용한 단방향 예측의 RD 비용값(J_{uni_0}), L_0 를 이용한 단방향 예측의 최적 움직임 벡터(a_{uni_0}) 및 최적 참조 프레임 인덱스(i_{uni_0})를 결정할 수 있다.

인코더는 L_1 를 사용하여 단방향 움직임 예측을 수행하고, RD 비용을 연산한다(120). 모드 $m = 1$ 은 L_1 를 사용하는 단방향 예측을 의미한다. 인코더는 $m = 1$ 에서 L_1 를 이용한 단방향 예측의 RD 비용값(J_{uni_1}), L_1 를 이용한 단방향 예측의 최적 움직임 벡터(a_{uni_1}) 및 최적 참조 프레임 인덱스(i_{uni_1})를 결정할 수 있다.

인코더는 L_0 및 L_1 를 사용하여 양방향 움직임 예측을 수행하고, RD 비용을 연산한다(130). 모드 $m = 2$ 은 L_0 및 L_1 을 사용하는 양방향 예측을 의미한다. 인코더는 $m = 2$ 에서 L_0 를 이용한 양방향 예측의 RD 비용값(J_{bi_0}), L_0 를 이용한 단방향 예측의 최적 움직임 벡터(a_{bi_0}) 및 최적 참조 프레임 인덱스(i_{bi_0})를 결정할 수 있다. 또 인코더는 $m = 2$ 에서 L_1 를 이용한 단방향 예측의 RD 비용값(J_{bi_1}), L_1 를 이용한 단방향 예측의 최적 움직임 벡터(a_{bi_1}) 및 최적 참조 프레임 인덱스(i_{bi_1})를 결정할 수 있다.

정규 움직임 예측은 영상 코딩 과정에서 사용되는 일반적인 움직임 예측 기법을 의미한다. 정규 움직임 예측 과정(110~130)을 제1 움직임 예측이라고 명명할 수 있다.

이후 인코더는 아핀 움직임 예측을 수행한다(140~150). 인코더는 4-파라미터 기반 아핀 움직임 예측을 수행한다(140). 모드 $m = 3$ 은 4-파라미터 기반 아핀 움직임 예측을 의미한다. 인코더는 $m = 3$ 에서 아핀 예측의 RD 비용값(J_{aff-4})를 결정할 수 있다. 인코더는 6-파라미터 기반 아핀 움직임 예측을 수행한다(150). 모드 $m = 4$ 은 6-파라미터 기반 아핀 움직임 예측을 의미한다. 인코더는 $m = 4$ 에서 아핀 예측의 RD 비용값(J_{aff-6})를 결정할 수 있다. 제1 움직임 예측과 구별하기 위하여 아핀 움직임 예측 과정(140~150)을 제2 움직임 예측이라고 명명할 수 있다.

한편 인코더는 아핀 움직임 예측에서도 L_0 를 사용한 단방향 예측, L_1 를 사용하여 단방향 예측 및 L_0 및 L_1 을 사용

한 양방향 예측을 수행한다. 4-파라미터 기반 아핀 움직임 예측(140) 과정에서 인코더는 L_0 를 사용한 단방향 예측(141), L_1 를 사용하여 단방향 예측(142) 및 L_0 및 L_1 를 사용한 양방향 예측(143)을 수행한다. 6-파라미터 기반 아핀 움직임 예측(150) 과정에서 인코더는 L_0 를 사용한 단방향 예측(151), L_1 를 사용하여 단방향 예측(152) 및 L_0 및 L_1 를 사용한 양방향 예측(153)을 수행한다. 도 3에 도시한 바와 같이, 4-파라미터 기반 아핀 움직임 예측(140) 및 6-파라미터 기반 아핀 움직임 예측(150) 과정에서, 인코더는 정규 움직임 예측(제1 움직임 예측) 과정과 유사하게 최적 움직임 벡터 a 및 최적 참조 프레임 인덱스 i 를 결정할 수 있다.

[0061] 최종적으로, 인코더는 모든 움직임 예측의 결과를 이용하여 해당 블록에 대한 최적 움직임 모드 m , 최적 움직임 벡터 a , 최적 참조 프레임 인덱스 i , 참조 프레임 리스트 φ 를 결정한다(160). 인코더는 RD 비용값 J 를 기준으로 가장 작은 값을 가진 모드를 최적 움직임 모드 m 로 선택하고, 선택한 모드에서 결정된 최적 움직임 벡터 a , 최적 참조 프레임 인덱스 i , 참조 프레임 리스트 φ 를 선택할 수 있다. 인코더는 해당 블록에 대한 최적 움직임 모드 및 관련 정보를 저장할 수 있다.

[0063] 이하 VVC 코딩 과정에서의 새로운 아핀 움직임 예측 기법에 대하여 설명한다. 새로운 아핀 움직임 예측 기법을 고속 아핀 움직임 예측 기법이라고 명명한다. 고속 아핀 움직임 예측은 아핀 움직임 예측에서 불필요할 수 있는 과정을 생략(skip)하여 움직임 예측의 복잡도를 낮춘다.

[0064] 고속 아핀 움직임 예측은 이전에 코딩된 정보로부터 추정되는 현재 CU의 움직임 특징을 반영한다. 간략하게 고속 아핀 움직임 예측의 특징을 먼저 설명한다.

[0065] 고속 아핀 움직임 예측은 두 가지 단계에서 복잡도를 낮출 수 있다. 첫째로, 고속 아핀 움직임 예측은 MTT에서 부모 CU 레벨에서 자식 CU의 움직임 예측을 초기에 종료할 수 있다(움직임 예측 생략). 즉, 인코더가 부모 CU의 최적 예측 모드의 정보를 이용하여 해당 부모 CU의 하위에 있는 모든 CU에 대한 움직임 예측을 생략할 수 있다.

[0066] 둘째로, 고속 아핀 움직임 예측은 정규 움직임 예측(제1 움직임 예측)에서 도출한 최적 참조 프레임의 움직임 예측 방향을 이용하여 현재 CU에 대한 참조 프레임 개수를 줄일 수 있다(참조 프레임 범위 한정).

[0068] 다른 프레임 간 예측 방법 중에서 아핀 움직임 예측이 최적일 확률을 사전 확률 $p(A)$ 라고 정의한다. 베이지 이론(Bayes theorem)에 따르면, 움직임 예측을 위하여 제안된 특징들이 나타나면 사후 확률(posterior probability)이 연산될 수 있다.

[0069] 부모 CU의 최적 예측 모드가 생략 모드일 경우의 확률을 $p(S_{par})$ 라고 정의한다. 생략 모드는 잔차 코딩없이, 병합 인덱스(merge index)만 필요하다. 생략 모드는 해당 블록에 매우 작은 움직임이 있는 경우라고 추정될 수 있다.

[0070] 정규 움직임 예측에서 최적 예측이 단방향 예측일 확률을 $p(U_{CME})$ 라고 정의한다. 장면(scene)들 사이에서 움직임이 변환적 움직임이거나 원거리 움직임인 경우, 단방향 예측이 선택된다. 단방향 예측은 줌인/줌아웃 또는 회전과 같은 비선형적 움직임의 예측에 취약하다. 따라서, 정규 움직임 예측에서 최적 모드로 단방향 움직임 예측이 선택되었다면, 해당 블록내의 움직임은 아핀 움직임 예측이 필요하지 않은 단순한 움직임이라고 추정될 수 있다.

[0072] 우도(likelihood) 확률 $p(S_{par}|A)$ 와 $p(S_{par})$ 를 수집하여, 아래의 수학식 6과 같이 사후 확률 $p(A|S_{par})$ 를 연산할 수 있다.

수학식 6

$$p(A | S_{par}) = \frac{p(S_{par} | A) p(A)}{p(S_{par})}$$

[0073]

[0074] 이와 유사하게, 아래 수학식 7과 같이 사후 확률 $p(A|U_{CME})$ 가 $p(U_{CME})$ 와 우도 확률 $p(U_{CME}|A)$ 를 관찰하여 연산될 수 있다.

수학식 7

$$p(A | U_{CME}) = \frac{p(U_{CME} | A) p(A)}{p(U_{CME})}$$

아래 표 1은 이전에 코딩된 CU의 움직임 데이터의 확률을 나타낸다. 표 1은 4개의 UHD 비디오 시퀀스들에 대한 확률을 나타낸다. 비디오 시퀀스들은 RA 설정에서 VTM 3.0이 코딩한 것이다. 비디오 시퀀스들은 단순히 25 및 35의 양자화 파라미터(quantization parameter, QP)를 사용하여 시퀀스 당 200 프레임으로 인코딩되었다.

표 1

Sequence	QP	$p(A)$	$p(S_{par})$	$p(A S_{par})$	$p(U_{CME})$	$p(A U_{CME})$
Bosphorus	25	27%	39%	12%	43%	24%
	35	18%	47%	8%	67%	14%
Jockey	25	29%	28%	10%	74%	25%
	35	22%	35%	8%	84%	19%
YachtRide	25	36%	23%	15%	50%	34%
	35	26%	34%	14%	64%	23%
HoneyBee	25	7%	78%	4%	89%	5%
	35	4%	79%	2%	96%	3%
Average		21%	45%	9%	71%	18%

표 1을 살펴보면, $p(A)$ 는 각 시퀀스마다 다르다. 이는 아핀 움직임 예측의 정확도가 비디오 시퀀스의 움직임 특징에 따라 쉽게 영향을 받는다는 것을 나타낸다. 그러나, 사전 확률과 비교하면, $p(A | S_{par})$ 는 값이 작지만 비교적 높은 값을 갖는다. 이는 현재 CU 이전의 부모 CU가 생략 모드인 경우, VTM 인코더가 많은 아핀 움직임 예측을 생략할 수 있다는 것을 시사한다. 예컨대, Bosphorus 시퀀스 및 Jockey 시퀀스에서 사전 확률은 29% 정도로 높지만, 사후 확률은 사전 확률에 비하여 매우 낮다.

다양한 비디오 특성에 따라, YachtRide와 같이 움직임에 민감한 시퀀스는 생략 모드를 선호하지 않는다. $p(S_{par})$ 가 40% 이하이다. 그러나 $p(U_{CME})$ 는 높다. $p(U_{CME})$ 는 YachtRide에서 64%, Jockey(QP=35)에서 84%를 나타낸다. 사후 확률 $p(A | U_{CME})$ 는 $p(A)$ 보다 작아진다.

부모 CU의 예측 모드 및 정규 움직임 예측의 예측 방향이라는 두 가지 조건을 고려하면, 아핀 움직임 예측이 불필요한 경우가 명확해 진다. 표 1에서 4개의 비디오 시퀀스의 평균 $p(A | S_{par})$ 는 9%이고, 평균 $p(A | U_{CME})$ 는 20% 이다.

전술한 바와 같이 고속 움직임 예측은 부모 CU의 코딩 정보를 이용하여 자식 CU의 예측 모드 생략 또는 참조 프레임 모집단의 범위를 줄여서 예측하는 과정을 포함한다.

도 4는 VVC에서 CU에 대한 고속 움직임 예측을 수행하는 과정(200)에 대한 예이다. 참조 프레임 리스트들은 L_0 및 L_1 이 사용된다고 전제한다. 부모 CU에 대한 자식 CU 형태(분할 구조)는 결정되었다고 가정한다. 도 4는 자식 CU 중 현재 움직임 예측을 수행하고자 하는 대상(타깃 CU라고 명명)에 대한 움직임 예측 과정이다. 부모 CU가 생략 모드가 아닌 경우, 각 자식 CU를 대상으로 움직임 예측을 수행할 수 있다.

인코더는 타깃 CU가 결정되기 전에, 타깃 CU의 부모 CU에 대한 움직임 예측 과정을 수행한다(210). 이 과정에서 인코더는 부모 CU에 대한 움직임 예측 정보를 획득한다. 부모 CU의 움직임 예측 정보는 부모 CU의 최종 움직임 예측 모드가 생략 모드(skip mode)인지 여부에 대한 정보를 포함한다.

고속 움직임 예측 과정도 타깃 CU에 대한 정규 움직임 예측은 수행한다. 인코더는 먼저 타깃 CU에 대한 정규 움

직업 예측을 수행한다(220~240).

- [0089] 인코더는 L_0 를 사용하여 단방향 움직임 예측을 수행하고, RD 비용을 연산한다(220). 모드 $m = 0$ 은 L_0 를 사용하는 단방향 예측을 의미한다. 인코더는 $m = 0$ 에서 L_0 를 이용한 단방향 예측의 RD 비용값(J_{uni_0}), L_0 를 이용한 단방향 예측의 최적 움직임 벡터(a_{uni_0}) 및 최적 참조 프레임 인덱스(I_{uni_0})를 결정할 수 있다.
- [0090] 인코더는 L_1 를 사용하여 단방향 움직임 예측을 수행하고, RD 비용을 연산한다(230). 모드 $m = 1$ 은 L_1 를 사용하는 단방향 예측을 의미한다. 인코더는 $m = 1$ 에서 L_1 를 이용한 단방향 예측의 RD 비용값(J_{uni_1}), L_1 를 이용한 단방향 예측의 최적 움직임 벡터(a_{uni_1}) 및 최적 참조 프레임 인덱스(I_{uni_1})를 결정할 수 있다.
- [0091] 인코더는 L_0 및 L_1 를 사용하여 양방향 움직임 예측을 수행하고, RD 비용을 연산한다(240). 모드 $m = 2$ 는 L_0 및 L_1 을 사용하는 양방향 예측을 의미한다. 인코더는 $m = 2$ 에서 L_0 를 이용한 양방향 예측의 RD 비용값(J_{bi_0}), L_0 를 이용한 단방향 예측의 최적 움직임 벡터(a_{bi_0}) 및 최적 참조 프레임 인덱스(I_{bi_0})를 결정할 수 있다. 또 인코더는 $m = 2$ 에서 L_1 를 이용한 단방향 예측의 RD 비용값(J_{bi_1}), L_1 를 이용한 단방향 예측의 최적 움직임 벡터(a_{bi_1}) 및 최적 참조 프레임 인덱스(I_{bi_1})를 결정할 수 있다.
- [0092] 정규 움직임 예측 과정(210~230)은 제1 움직임 예측이라고 명명할 수 있다.
- [0093] 인코더는 부모 CU의 예측 모드가 생략 모드인지를 확인한다(250). 부모 CU의 예측 모드가 생략 모드인 경우(250의 YES), 타깃 CU에 대한 아핀 예측을 생략한다. 이 경우 인코더는 정규 예측 과정에서 도출된 정보만을 사용하여 최적 예측 모드를 결정한다(290). 부모 CU의 예측 모드가 생략 모드인 경우, 해당 부모 CU의 하위에 있는 CU에 대한 아핀 예측은 모두 생략될 수 있다.
- [0094] 부모 CU의 예측 모드가 생략 모드가 아닌 경우(250의 NO), 인코더는 타깃 CU에 대한 아핀 예측을 수행한다. 이때 인코더는 타깃 CU에 대한 정규 움직임 예측 과정에서 산출한 정보를 이용하여, 타깃 CU의 아핀 예측에서 사용할 참조 프레임 범위를 한정할 수 있다. 정규 움직임 예측 과정에서 산출한 정보를 정규 움직임 예측 정보라고 명명한다. 정규 움직임 예측 정보는 (i) $m = 0$ 에서의 RD 비용값(J_{uni_0}), 최적 참조 프레임 인덱스(I_{uni_0}), (ii) $m = 1$ 에서의 RD 비용값(J_{uni_1}), 최적 참조 프레임 인덱스(I_{uni_1}) 및 (iii) $m = 2$ 에서의 RD 비용값(J_{bi})을 포함할 수 있다.
- [0095] 인코더는 타깃 CU에 대한 정규 움직임 예측 정보 중 RD 비용값을 비교하여 아핀 예측을 위한 참조 프레임 범위를 한정할 수 있다.
- [0096] 인코더는 L_0 를 사용한 단방향 예측($m = 0$)의 RD 비용값(J_{uni_0})과 L_1 를 사용한 단방향 예측($m = 1$)의 RD 비용값(J_{uni_1})을 비교할 수 있다(261). 또 인코더는 L_0 를 사용한 단방향 예측($m = 0$)의 RD 비용값(J_{uni_0})과 양방향 예측($m = 2$)의 RD 비용값(J_{bi})을 비교할 수 있다(261).
- [0097] 인코더는 $J_{uni_0} < J_{uni_1}$ 이고, 동시에 $J_{uni_0} < J_{bi}$ 인 경우(261의 YES), 참조 프레임의 범위를 한정할 수 있다(262). $J_{uni_0} < J_{uni_1}$ 이고, 동시에 $J_{uni_0} < J_{bi}$ 인 경우 인코더는 아핀 예측에서 사용할 L_0 의 범위를 한정할 수 있다. $m = 0$ 이면(최적 참조 프레임 리스트가 L_0 인 경우), 아핀 예측에서 사용하는 L_0 의 인덱스 범위를 $\{0 \sim I_{uni_0}\}$ 로 설정할 수 있다(262). 이 경우, L_1 의 범위는 축소하지 않는다.
- [0098] 인코더는 L_0 를 사용한 단방향 예측($m = 0$)의 RD 비용값(J_{uni_0})과 L_1 를 사용한 단방향 예측($m = 1$)의 RD 비용값(J_{uni_1})을 비교할 수 있다(263). J_{uni_0} 과 J_{uni_1} 비교는 261 과정에서 수행한 결과를 이용할 수 있다. 또 인코더는 L_1 를 사용한 단방향 예측($m = 1$)의 RD 비용값(J_{uni_1})과 양방향 예측($m = 2$)의 RD 비용값(J_{bi})을 비교할 수 있다(263).
- [0099] 인코더는 $J_{uni_0} > J_{uni_1}$ 이고, 동시에 $J_{uni_1} < J_{bi}$ 인 경우, 참조 프레임의 범위를 한정할 수 있다(264). $J_{uni_0} > J_{uni_1}$ 이고, 동시에 $J_{uni_1} < J_{bi}$ 인 경우, 인코더는 아핀 예측에서 사용할 L_1 의 범위를 한정할 수 있다. $m = 1$ 이면(최적 참조 프레임 리스트가 L_1 인 경우), 아핀 예측에서 사용하는 L_1 의 인덱스 범위를 $\{0 \sim I_{uni_1}\}$ 로 설정할 수

있다. 이 경우, L_0 의 범위는 축소하지 않는다.

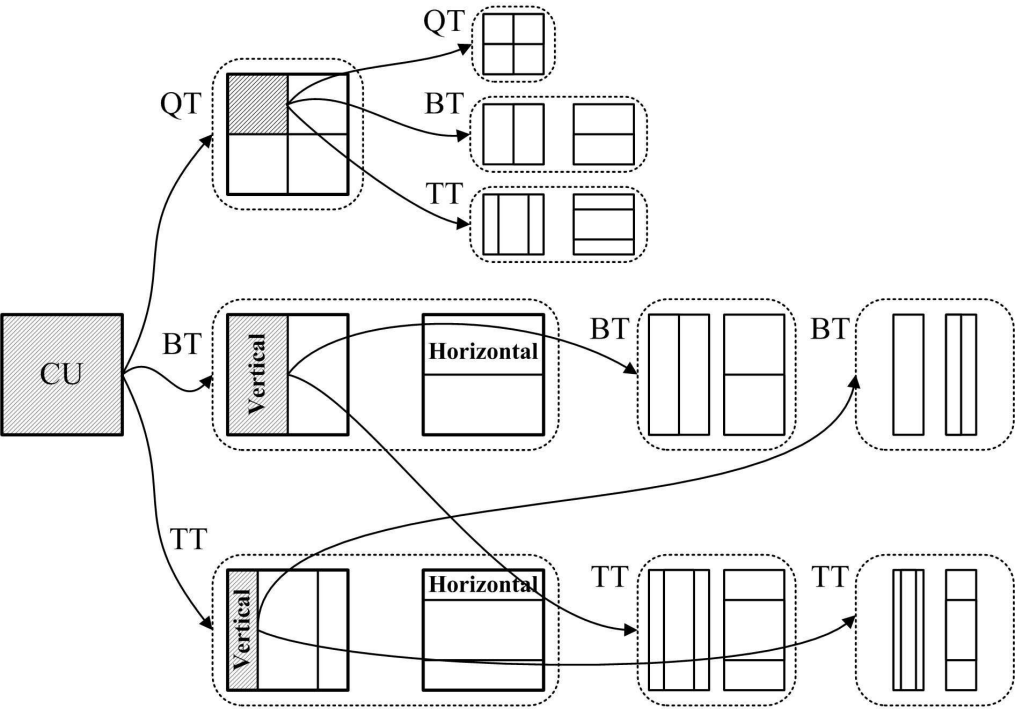
- [0100] 이후 인코더는 4-파라미터 기반 아핀 움직임 예측을 수행한다(270). 모드 $m = 3$ 은 4-파라미터 기반 아핀 움직임 예측을 의미한다. 인코더는 $m = 3$ 에서 아핀 예측의 RD 비용값(J_{aff-4})를 결정할 수 있다. 인코더는 6-파라미터 기반 아핀 움직임 예측을 수행한다(280). 모드 $m = 4$ 은 6-파라미터 기반 아핀 움직임 예측을 의미한다. 인코더는 $m = 4$ 에서 아핀 예측의 RD 비용값(J_{aff-6})를 결정할 수 있다. 제1 움직임 예측과 구별하기 위하여 아핀 움직임 예측 과정(270~280)을 제2 움직임 예측이라고 명명할 수 있다.
- [0101] 인코더는 아핀 움직임 예측에서도 L_0 를 사용한 단방향 예측, L_1 을 사용하여 단방향 예측 및 L_0 및 L_1 을 사용한 양방향 예측을 수행한다. 이때 아핀 움직임 예측에 사용할 참조 프레임 범위가 한정된 경우(262 또는 264), 인코더는 한정된 참조 프레임을 사용하여 아핀 예측을 수행한다.
- [0102] 따라서, $m = 2$ 이면, 인코더는 종래(도 3) 방식에 따라 아핀 예측을 수행할 수 있다. 즉, 인코더는 참조 프레임 범위를 한정하지 않고 아핀 예측을 수행할 수 있다.
- [0103] 최종적으로, 인코더는 움직임 예측의 결과를 이용하여 타깃 CU에 대한 최적 움직임 모드 m , 최적 움직임 벡터 a , 최적 참조 프레임 인덱스 i , 참조 프레임 리스트 ϕ 를 결정한다(290). 인코더는 해당 블록에 대한 최적 움직임 모드 및 관련 정보를 저장할 수 있다.
- [0105] 도 5는 고속 움직임 예측 장치의 구성(300)을 도시한 예이다. 움직임 예측 장치(300)는 영상에 대한 움직임 예측을 수행하는 인코더의 내부 컴포넌트일 수 있다. 움직임 예측 장치(300)는 아날로그 회로 또는 디지털 회로로 구현될 수 있다. 도 5는 움직임 예측을 위한 코드가 임베드된 장치를 예로 도시한다. 도 5는 움직임 예측을 위한 코드 내지 프로그램이 임베드된 칩과 같은 형태의 움직임 예측 장치(330)를 도시한다. 움직임 예측 장치(330)는 VVC 표준에 따르는 영상에 대한 움직임 예측을 수행할 수 있다.
- [0106] 움직임 예측 장치(300)는 저장 장치(310), 프로세서(320) 및 인터페이스 장치(330)를 포함한다.
- [0107] 저장 장치(310)는 움직임 예측에 필요한 정보를 저장한다. 저장 장치(310)는 움직임 예측에 필요한 정보를 임시 저장할 수 있다. 이 경우 저장 장치(310)는 플래시 메모리와 같은 장치일 수 있다. 나아가 저장 장치(310)는 움직임 예측을 위한 코드 내지 프로그램을 저장할 수 있다. 이 경우 저장 장치(310)는 ROM과 같은 장치일 수 있다. 따라서, 도 5에서 저장 장치(310)로 하나의 블록을 도시하였지만, 저장 장치(310)는 임시 정보를 저장하는 메모리와 움직임 예측 코드를 저장하는 저장 매체로 구분될 수도 있다.
- [0108] 저장 장치(310)는 움직임 예측에 필요한 다양한 정보를 저장할 수 있다. 저장 장치(310)는 프레임 정보, 블록 정보, 블록 분할 정보 등과 같은 정보를 수신하여 저장할 수 있다. 블록 분할 정보는 MTT에 기반하여 분할된 블록에 대한 정보를 포함할 수 있다. 블록 분할 정보는 부모 CU와 상기 부모 CU의 자식 CU에 대한 구조 및 정보를 포함할 수 있다.
- [0109] 나아가 저장 장치(310)는 이전 움직임 예측을 수행한 결과로 산출되는 정보를 피드백받아 저장할 수 있다. 저장 장치(310)는 부모 블록(상위 블록) 또는 부모 CU의 움직임 예측을 수행한 결과를 저장할 수 있다. 예컨대, 저장 장치(310)는 부모 CU에 대한 움직임 예측 정보를 저장할 수 있다. 부모 CU의 움직임 예측 정보는 부모 CU에 대한 최종 예측 모드가 생략 모드인지에 대한 정보를 포함한다. 또 저장 장치(310)는 현재 움직임 예측 대상인 블록(하위 블록)에 대한 정규 움직임 예측 정보를 저장할 수 있다. 예컨대, 저장 장치(310)는 타깃 CU에 대한 (i) $m = 0$ 에서의 RD 비용값(J_{uni-0}), 최적 참조 프레임 인덱스(I_{uni-0}), (ii) $m = 1$ 에서의 RD 비용값(J_{uni-1}), 최적 참조 프레임 인덱스(I_{uni-1}) 및 (iii) $m = 2$ 에서의 RD 비용값(J_{bi})을 저장할 수 있다.
- [0110] 인터페이스 장치(330)는 인코더 내부에서 데이터 및 명령을 전달하는 구성을 의미한다. 인터페이스 장치(330)는 내부 통신을 위한 물리적 장치 및 통신 프로토콜을 포함할 수 있다. 인터페이스 장치(330)는 정보를 전달하는 회선, 정보 전달을 제어하는 구성(스위치 등), 정보를 가공하는 구성(합산기 등) 등으로 구성될 수 있다.
- [0111] 프로세서(320)는 주어진 데이터 내지 정보를 처리하는 구성을 의미한다. 프로세서(320)는 일종의 연산 장치이다. 프로세서(320)는 저장 장치(310)에 저장된 코드 내지 프로그램을 이용하여 영상에 대한 움직임 예측을 수행할 수 있다.
- [0112] 프로세서(320)는 특정 블록 내지 CU에 대한 움직임 예측을 수행한다. 예컨대, 프로세서(320)는 CU에 대한 정규 움직임 예측(전술한 제1 움직임 예측) 및 아핀 움직임 예측(전술한 제2 움직임 예측)을 수행할 수 있다. 프로세

서(320)는 블록에 대한 분할도 수행할 수 있다. 예컨대, 프로세서(320)는 QT(quad-tree), TT(ternary tree) 및 BT(binary tree) 타입을 포함하는 MTT(multi-type tree) 구조를 기준으로 특정 블록을 분할할 수 있다.

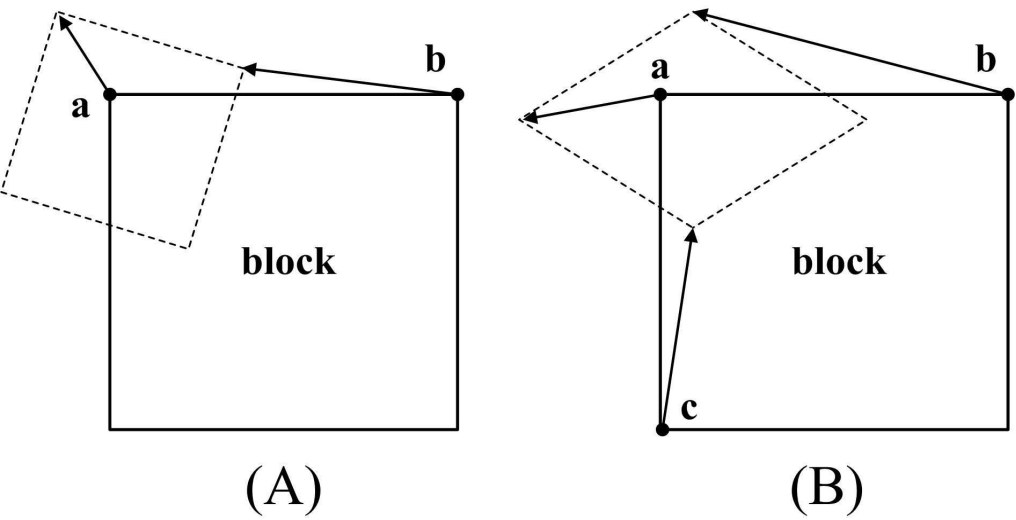
- [0113] 프로세서(320)는 상위 블록 또는 부모 CU에 대한 움직임 예측을 수행한다. 도 5에서 도시한 바와 같이, 상위 블록 또는 부모 CU에 대한 움직임 예측 정보는 하위 레벨의 움직임 예측을 위하여 저장할 수 있다. 프로세서(320)는 하위 블록 또는 자식 CU에 대한 움직임 예측을 수행한다. 이때, 프로세서(320)는 상위 블록 또는 부모 CU에 대한 움직임 예측 정보를 활용하여 하위 블록 또는 자식 CU에 대한 움직임 예측을 수행할 수 있다. 움직임 예측 과정은 정규 움직임 예측과 아핀 움직임 예측을 포함한다.
- [0114] 예컨대, 프로세서(320)는 상위 블록(또는 부모 CU)의 움직임 예측이 생략된 경우, 하위 블록(또는 자식 CU)에 대한 아핀 움직임 예측을 생략할 수 있다. 프로세서(320)는 상위 블록(또는 부모 CU)의 움직임 예측 수행된 경우, 하위 블록(또는 자식 CU)에 대한 정규 움직임 예측 정보를 기준으로 아핀 움직임 예측에 사용할 참조 프레임 범위를 한정할 수 있다.
- [0115] 프로세서(320)는 $J_{uni_0} < J_{uni_1}$ 이고 $J_{uni_0} < J_{bi}$ 인 경우, 아핀 예측에서 L_0 의 범위를 '인덱스 0 ~ I_{uni_0} ' 범위로 한정하여 하위 블록(또는 자식 CU)에 대한 아핀 예측을 수행할 수 있다.
- [0116] 프로세서(320)는 $J_{uni_0} > J_{uni_1}$ 이고 $J_{uni_1} < J_{bi}$ 인 경우, 아핀 예측에서 L_1 의 범위를 '인덱스 0 ~ I_{uni_1} ' 범위로 한정하여 하위 블록(또는 자식 CU)에 대한 아핀 예측을 수행할 수 있다.
- [0118] 또한, 상술한 바와 같은 고속 움직임 예측 방법은 컴퓨터에서 실행될 수 있는 실행가능한 알고리즘을 포함하는 프로그램(또는 어플리케이션)으로 구현될 수 있다. 상기 프로그램은 비일시적 판독 가능 매체(non-transitory computer readable medium)에 저장되어 제공될 수 있다.
- [0119] 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.
- [0121] 본 실시예 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시예는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

도면

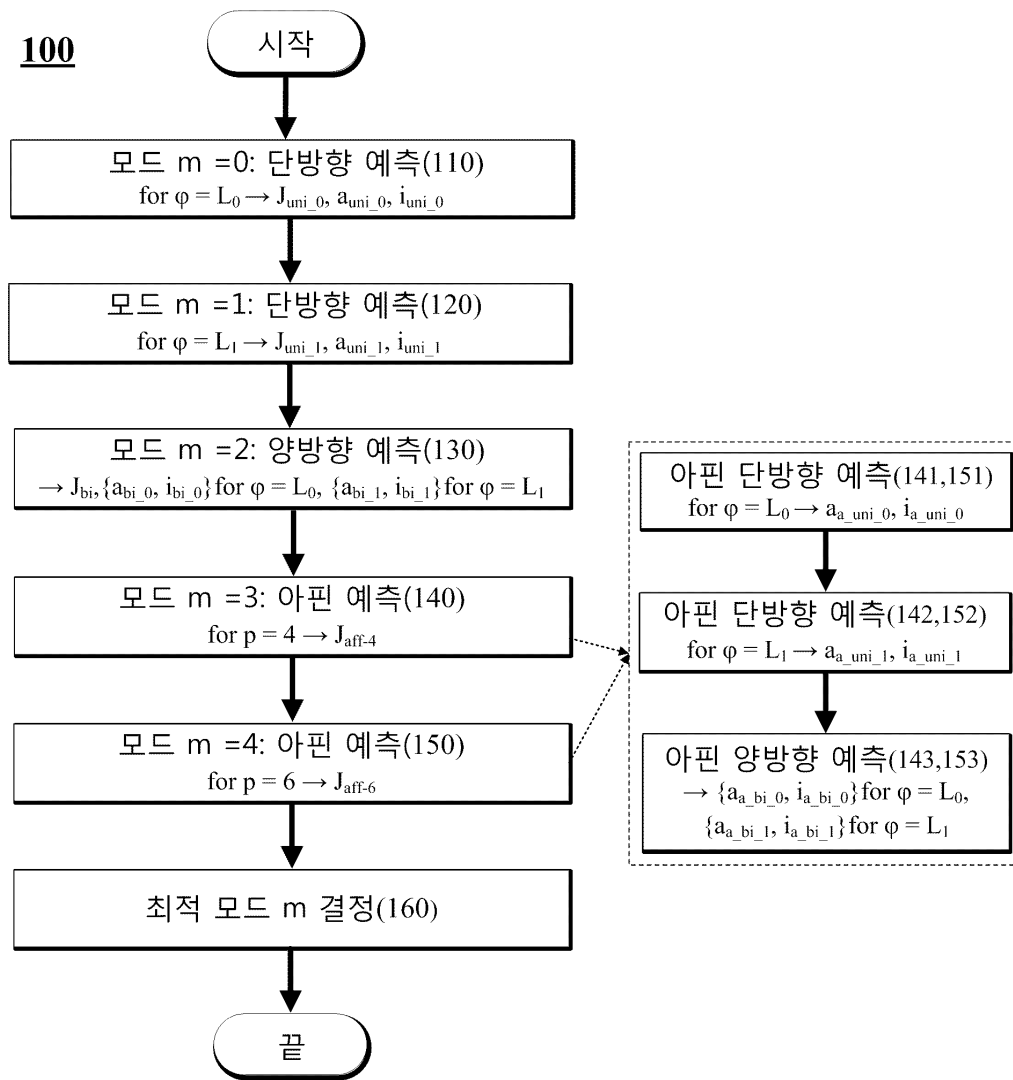
도면1



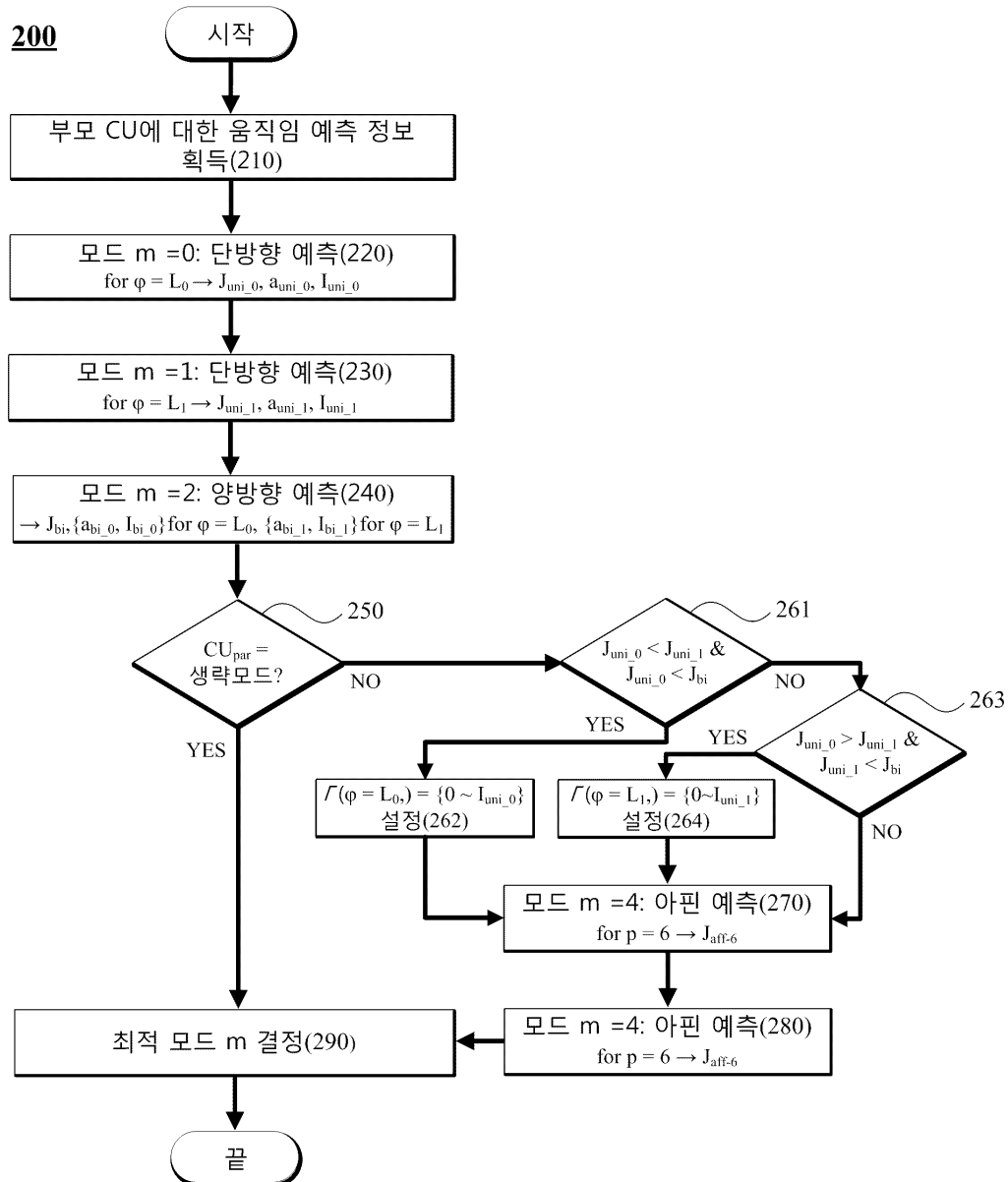
도면2



도면3



도면4



도면5

